

Joomla! Sicherheits Checkliste

Freitag, 10. November 2006

Dies ist eine (freie, damit hoffentlich verständlichere) Übersetzung der Joomla! Sicherheits Checkliste.

Original: forum.joomla.org

Gefunden auf: www.joomlaportal.de

Der Dank für die Übersetzung geht an:

Anne-Kathrin Merz aka "rabe"

Joomla User Group Regensburg: www.jug-regensburg.de

Wir haben uns bemüht, die Übersetzung so zu gestalten, dass Sie jederzeit leicht entsprechend des englischsprachigen Threads aktualisiert werden kann, allerdings einige Änderungen vorgenommen:

- Zum besseren Verständnis wird der Text nach den jeweiligen Möglichkeiten, die ein User auf einem Shared- bzw. einem Root-Server hat, unterteilt.
- Links auf kommerzielle Anbieter wurden entfernt, entsprechende Links sind jetzt, soweit möglich auf deutschsprachige Suche angepasst
- Einige Anmerkungen entsprechen der Auffassung der deutschen Übersetzer

Wichtige Information: Bitte zuerst lesen!

- Was bietet dieser Artikel: Einen schnellen Überblick über Joomla! Sicherheitsfragen mit vielen Links auf weiterführende Information.

- Was bietet dieser Artikel nicht: Eine komplette Antwortliste auf alle möglichen Sicherheitsfragen, einen Ersatz für jahrelange harte Arbeit, gründliche Studien und Erfahrung. Internetsuche, kreativ eingesetzt, wird Sie weit bringen. Zum Beispiel gibt es wiederholt Fragen zu SSH. Eine Google Suche zum Thema "SSH" liefert viele gute Links. Hier ist einer davon: [www.google.de/search?...](http://www.google.de/search?)

- Wie Sie helfen können: Joomla! ist ein Community Projekt und wird nach Bedarf aktualisiert. Wenn Sie Fehler finden, teilen Sie uns diese bitte mit. Korrekturen sind willkommen und können in folgendem Topic des offiziellen Forums auf Joomla.org eingestellt werden:

Topic: Discussion for: Joomla Administrator's Security Checklist forum.joomla.org/index.php/topic,81959.0.html

Oder direkt hier in diesem Forum im Thread Diskussion zur Joomla! Sicherheits Checkliste: www.joomlaportal.de

Allgemeine Überlegungen

- • PHP, MySQL und viele andere Basis Komponenten wurden ursprünglich rein für Apache Server konzipiert, wo sie auch am Besten funktionieren. Vermeiden Sie andere Server, wenn es möglich ist.
- • Ändern Sie regelmäßig Passwörter und vergeben Sie diese eindeutig. Benutzen Sie dafür am Besten eine zufällige Kombination aus Buchstaben, Zahlen oder Symbolen und vermeiden Sie es, einzelne Namen oder Wörter zu verwenden, die auch in einem Wörterbuch gelistet sind. Benutzen Sie nie die Namen von Verwandten, Haustieren und ähnlichem.
- • Wenn Sie einen Shared Hosting Provider nutzen, stellen Sie bitte sicher, dass keine anderen Benutzer auf Ihrem Server Ihre Dateien sehen können, beispielsweise durch Shell Accounts oder CPanel.
- • Verlassen Sie sich nie auf Backups anderer. Tragen Sie persönlich die Verantwortung für die Datensicherung von Dateien und Datenbank. Viele Serviceprovider sichern vertraglich ab, dass Sie sich nicht auf deren Sicherungsdaten verlassen können.
- • Benutzen Sie Systeme, mit denen Sie das Eindringen Fremder in Form bössartiger HTTP Anfragen verhindern und entdecken können.

Entwicklungsserver

- • Setzen Sie einen lokalen Server für die Entwicklung auf, nehmen Sie auch Upgrades zunächst hier vor und testen Sie diese, bevor Sie das System auf dem Produktionsserver einspielen. Apache Friends stellt XAMPP zur Verfügung, ein frei verfügbare LAMP Installation, die auf vielen Betriebssystemen, inklusive GNU/Linux und Windows, funktioniert.
- • Hier können Sie XAMPP herunterladen <http://www.apachefriends.org/de/index.html>
- • Nutzen Sie für eine schnellere und professionellere Entwicklung eine sog. IDE
- • Einige ISPs bieten Entwicklungsserver inklusive Backup an.

HTTP Server

- • Nutzen Sie die Einstellungsmöglichkeiten der .htaccess Datei, um sich vor Angriffen von außen zu schützen. Ein sehr gutes Tutorial finden Sie hier:
- • Topic: Using .htaccess files to block exploit attempts: <http://forum.joomla.org/index.php/topic,75376.0.html>
- • Für Joomla! 1.0.11 sind diese Einstellungen Teil der Installationsroutine.
- • Kontrollieren Sie regelmäßig die Server Log Dateien und überprüfen Sie diese auf verdächtige Zugriffe. Dabei sollten Sie sich nicht auf Zusammenfassungen oder graphische Darstellungen verlassen. Kontrollieren Sie also die blanken Log Dateien.

MYSQL

- • Wenn Sie einen Shared Server nutzen und die Namen anderer Datenbanken aufgelistet sehen können, können Sie sicher sein, dass auch andere Benutzer Ihre Daten sehen. Dies ist ein unnötiger Schritt näher dahin, dass jemand in Ihre Datenbank eindringen kann. Ein guter Service Provider wird immer dafür sorgen, dass jeder Benutzer nur seine eigene Datenbank sehen kann.

PHP

- • Zum jetzigen Zeitpunkt werden PHP4 und PHP5 aktiv unterstützt. Stellen Sie sicher, dass Ihr gesamter PHP Code PHP5 kompatibel ist, bevor der Zeitpunkt gekommen ist, an dem PHP4 obsolet geworden ist. [Anmerkung: die Übersetzer sind sich bewusst, dass dies bei vielen Komponenten schlicht ein Zeitproblem ist]
- • PHP News <http://www.php.net/> (Anmerkung: Sie können Ihre persönliche Spracheinstellung vornehmen!)

Php.ini

- • Auf vielen Shared Hosting Systemen, bei denen Sie selbst keinen Zugriff auf die eigentliche php.ini Datei haben, dürfen Sie eigene php.ini Dateien erstellen. In den meisten Fällen müssen Sie dazu Ihre php.ini Datei in alle Unterverzeichnisse Ihres Webspaces kopieren. Falls das ein bisschen mühsam klingt: Es gibt einen frei verfügbaren Satz von Skripten, der diese Arbeit automatisch übernimmt. Das erste Skript kopiert die eigentliche php.ini in Ihr Hauptverzeichnis. Dieses kann dann getestet und angepasst werden. Das zweite Skript kopiert die neue php.ini vom Homeverzeichnis in alle Unterverzeichnisse (und überschreibt existierende php.ini Dateien). Hier ein Link zum Thema: Topic: secure it with php.ini <http://forum.joomla.org/index.php/to...html#msg411018>

Joomla!Core

- • Informieren Sie sich selbst über Joomla! 1.5. Dieses neue Release beinhaltet einige Verbesserungen.
- • Joomla! 1.5 Entwickler Status <http://dev.joomla.org/content/view/1139/82/>
- • Machen Sie immer einen Upgrade auf die aktuellste stabile Version.

- • Topic: How to patch Joomla! 1.0.x to 1.0.x? <http://forum.joomla.org/index.php/topic,33226.0.html>
- • Laden Sie Joomla! nur von den offiziellen und vertrauenswürdigen Seiten herunter, z.B. <http://forge.joomla.org/sf/sfmain/do...rojects.joomla>
- • Abonnieren Sie die Joomla! Sicherheitsankündigungen des offiziellen Joomla! Forums oder lesen Sie diese regelmäßig [Anmerkungen: dann sind Sie hier gerade richtig oder auch auf dem offiziellen Joomla Forum <http://forum.joomla.org/>]
- • Wenn Sie glauben, ein Sicherheitsrisiko entdeckt zu haben, so berichten Sie dieses bitte an das Entwicklerteam: Wie das geht, können Sie hier nachlesen: <http://dev.joomla.org/content/view/1450/89/>.
- • Entfernen Sie alle Templates, die für Ihre Installation nicht benötigt werden. Vor allem packen Sie keinerlei sicherheitsrelevanten Code in Ihre Template Dateien Mehr Information dazu hier <http://forum.joomla.org/index.php/to...html#msg430051> im offiziellen Joomla! Forum
- • Editieren Sie die Datei globals.php, um die Joomla! register_globals Emulation zu deaktivieren. Auch wenn diese Emulation sicherer ist als die PHP register_globals Anweisung, ist es immer noch am Besten, register_globals generell zu deaktivieren. Ab PHP6 werden Sie sich dies nicht mehr aussuchen können und es ist jetzt schon Zeit dafür. Hier die korrekten Einstellungen für die Joomla! register_globals Emulation:

PHP-Code: `define('RG_EMULATION', 0);`

- • Bitte beachten Sie jedoch, dass einige Erweiterungen eventuell nicht mehr richtig funktionieren, nachdem Sie diese Einstellung vorgenommen haben.

- • Wenn Ihre Seite konfiguriert ist und stabil läuft, versehen Sie so viele Verzeichnisse und Dateien wie möglich mit einem Schreibschutz, indem Sie die Rechte von 755 auf 644 zurücksetzen. [Anmerkung: bitte überprüfen Sie vorher, ob Ihre Joomla! Installation dann noch funktioniert, diese Einstellung ist Server-abhängig]
- • Es gibt unter den Einstellungen Site->GlobalConfiguration->Server ein Feature, das alle Ihre Verzeichnisse und Dateien auf den von Ihnen gewünschten Rechtstatus setzt [Anmerkung: bitte beachten Sie, dass dies nicht bei allen Servern möglich ist]. Bitte beachten Sie die Warnung: einige Drittanbieter Erweiterungen könnten mit dieser Einstellung nicht mehr richtig funktionieren. Testen Sie also vorher.
- • Hier einige Links auf die Anmerkungen des Autors der Originalversion dieses Artikels. Es geht dabei um eine Erweiterung, deren Konfigurationsfile sich mit dieser Einstellung nicht entsprechend setzen lässt: <http://help.joomla.org/content/view/41/132/>
<http://forum.joomla.org/index.php/topic,24108.0.html>
<http://forum.joomla.org/index.php?topic=87719>
- • Achtung: Wenn Sie irgendwann weitere Erweiterungen installieren möchten, müssen Sie die Berechtigungen zurücksetzen. Bitte beachten Sie auch, dass die Checkbox zum Überschreiben des Schreibschutzes der configuration.php Datei auf manchen Servern ausgegraut ist. Sie müssen in diesem Fall die Berechtigungen manuell setzen und natürlich auch nach den Anpassungen wieder zurück.

Joomla! Erweiterungen (Komponenten, Module und Bots)

- • Entfernen Sie alle Joomla! Erweiterungen, die register_globals ON erfordern
- • Laden Sie Ihre Erweiterungen nur von vertrauenswürdigen Seiten herunter. Die offizielle Definition für vertrauenswürdig ist dabei: Seiten, denen Sie selbst vertrauen
- • Vor der Installation einer Erweiterung sollten Sie die offizielle Liste der unsicheren Drittanbieter Erweiterungen lesen: <http://forum.joomla.org/index.php/topic,79477.0.html>
- • Liebe Benutzer, Achtung! Drittanbieter Erweiterungen gibt es in jeder nur erdenklichen Qualität und in jedem nur erdenklichen Alter. Auch wenn es den Joomla! Entwickler und Programmierstandard gibt, so werden die Erweiterungen, die auf der offiziellen Joomla! Seite gelistet sind keiner Qualitätsprüfung unterzogen. Testen Sie also alle Erweiterungen, bevor Sie damit auf eine Produktionsseite gehen.
- • Machen Sie regelmäßige Backups Ihrer FTP-Daten und auch der Datenbank, insbesondere bevor Sie neue Erweiterungen installieren.
- • Informieren Sie sich regelmäßig die Sicherheitsthemen auf der offiziellen Joomla! Seite unter <http://forum.joomla.org/index.php/board,296.0.html>
- • Entfernen Sie alle unbenutzten Joomla! Erweiterungen und kontrollieren Sie insbesondere auch, ob entsprechende Verzeichnisse und Dateien gelöscht wurden.

Wiederherstellung nach Hackerangriff!?

- • Wenn Ihre Seite gehackt wurde, stellen Sie sicher, ob sie wirklich gehackt wurde (genau!) und lesen Sie hier <http://forum.joomla.org/index.php/to...html#msg289697>
- • Bevor Sie irgendwelche Änderungen vornehmen, machen Sie eine Liste der Dateien sortiert nach Änderungsdatum
- • Es gibt ein Skript, das dies für Sie erledigt: <http://www.joomlation.eu/index.php?o...17&l+temid=35>
- • Kontrollieren Sie die originalen Logdateien auf Ihrem Server hinsichtlich verdächtiger Anfragen und Zugriffe

- Stellen Sie sicher, dass Sie alle Dateien und Verzeichnisse gelöscht haben, auch alle Unterverzeichnisse und alle Dateien sowie die Datenbank Tabellen.

- Dann erst installieren Sie alles neu, auf Basis Ihrer Backup Dateien.

Für User auf Root-Servern http-Server:

- Konfigurieren Sie die Apache Module `mod_security` und `mod_rewrite`, um PHP Angriffe zu verhindern.

MySQL:

- Stellen Sie sicher, dass der Joomla! MySQL Account nur eingeschränkte Rechte besitzt. Die Standardeinstellungen von MySQL sind unsicher, eine sorgfältige Konfiguration ist daher dringend erforderlich.

- Näheres dazu in der MySQL Dokumentation <http://dev.mysql.com/doc/refman/4.1/...rivileges.html>

PHP

- Spielen Sie alle von den Anbietern offiziell veröffentlichten Sicherheits-Patches so bald wie möglich ein.

- Halten Sie Ihr PHP serverseitig aktuell

- Benutzen Sie Werkzeuge, um automatisierte SQL Angriffe gegen PHP Anwendungen zu simulieren.

- Nähere Informationen unter Wikipedia: SQL-Injektion <http://de.wikipedia.org/wiki/SQL-Injektion>

- Folgen Sie dem Prinzip des "geringsten Zugriffsrechts" auf Ihr PHP, indem Sie Werkzeuge wie `PHPsuExec`, `php_suexec` oder `suPHP` einsetzen

`php.ini`

- Es gibt viele Methoden, Ihre Website durch die Konfiguration der `php.ini` sicherer zu machen. Der folgender Artikel (geschrieben von Beat, Q&T Workgroup Member) gibt Ihnen einen Überblick über die einzelnen Methoden sortiert nach der Reihenfolge, in der sie angewandt werden sollten: Sichern Sie Ihre Site mit `php.ini`

<http://forum.joomla.org/index.php/to...html#msg455771>

- Arbeiten Sie die Liste der `php.ini` Anweisungen auf der offiziellen PHP Website durch. <http://php.net/>

- Setzen Sie `register_globals` auf OFF: Diese Anweisung definiert, ob die EGPCS (Environment, GET, POST, Cookie, Server) Variablen als globale Variablen registriert werden oder nicht.

- Nutzen Sie `disable_functions`, um gefährliche PHP Funktionen, die von Ihrer Seite nicht benötigt werden, auszuschalten.

- Deaktivieren Sie `allow_url_open`. Diese Funktion ermöglicht es fopen Wrappern, auf URL Objekte wie Dateien zuzugreifen. Standard Wrapper werden für den Zugriff auf Remote Dateien, die das FTP oder HTTP Protokoll nutzen, eingesetzt. Einige Erweiterungen wie beispielsweise `zlib` können weitere Wrapper registrieren. Bitte beachten Sie: Dies kann aus Sicherheitsgründen ausschließlich in der `php.ini` gesetzt werden. [Anmerkung: mehr zu `fsock open` auf der `php.net` Seite als Alternative zu `fopen`: <http://de.php.net/fsockopen>]

- Passen Sie die `magic_quotes_gpc` Anweisung an Ihre Bedürfnisse an. Sie sollte auf OFF gesetzt sein, wenn Sie sich sicher sind, dass alle PHP Dateien auf Ihrem Server fehlerfrei programmiert sind und Benutzerdaten korrekt interpretiert werden. Sie sollte auf ON gesetzt sein, wenn Sie alte PHP3 und PHP4 Skripts laufen haben. Dazu gehören beispielsweise zu viele der Drittanbieter Komponenten. Joomla! 1.0.11 benachrichtigt Sie im Backend, wenn `magic_quotes_gpc` auf OFF gesetzt ist. Diese Sicherheitsmaßnahme wurde von den Joomla! Entwicklern zum Schutz vor Angriffen auf unsichere Drittanbieter Komponenten eingeführt. Das Joomla! Framework an sich wird in jedem Fall so weit wie möglich sicher gehalten.

Die Hintergründe zu `magic_quotes_gpc`. Was passiert?

`magic_quotes_gpc` setzt den `magic_quotes` Status für GPC(Get/Post/Cookie) Operationen. Falls `magic_quotes_gpc` auf ON gesetzt ist, werden alle Einfach Anführungsstriche, Anführungsstriche und Backslashes sowie Nullen automatisch durch einen führenden Backslash ergänzt.

Einige erfahrene Benutzer befürworten `magic_quotes_gpc ON` als zusätzliche Sicherheitsmaßnahme. Andererseits weist das offizielle PHP Handbuch auf folgendes hin:

"Die Einstellung `magic_quotes_gpc OFF` ist deutlich vorzuziehen, stattdessen sollten die Daten, falls nötig, zur Laufzeit mit einem Escape Character versehen werden".

PHP6 wird im Übrigen nur mit der Einstellung `magic_quotes_gpc OFF` lauffähig sein.

Da es in Zukunft einige technische Gründe geben wird, die die Einstellung `magic_quotes_gpc` immer wichtiger machen, könnten Sie jetzt bereits damit anfangen, Ihre Seiten entsprechend anzupassen.

- Wenn Ihre Seite einmal vollständig konfiguriert ist, können Sie damit beginnen, `safe_mode` zu aktivieren und korrekt zu konfigurieren. Beachten Sie bitte, dass `safe_mode ON` das Joomla! Installationskript beeinträchtigt. Für weitere Installationen kann `safe_mode` zeitweise ausgeschaltet werden, vergessen Sie aber nicht, ihn später wieder zu reaktivieren.

- Beachten Sie, dass einige erfahrene Benutzer der Meinung sind, ein aktivierter `safe_mode` sei nicht nötig, falls andere sicherheitsrelevante Überlegungen vorgenommen werden:

PHP Sicherheits- und Safe Mode Konfigurationsanweisungen

<http://us3.php.net/manual/de/feature...#ini.safe-mode>

PHP Funktionen, die mit safe_mode OFF nur eingeschränkt oder überhaupt nicht funktionieren

<http://us3.php.net/manual/en/feature....functions.php> [Anmerkung: dieser Link ist offenbar nur englisch verfügbar]

• open_basedir (dies sollte aktiviert und korrekt konfiguriert sein)

Dies beschränkt den Zugriff durch PHP auf spezielle Verzeichnisse wie auch auf die Dateien dieses Verzeichnisses.

Diese Einstellung hat nichts damit zu tun, ob safe_mode aktiviert oder deaktiviert ist. Bei der Beschränkung des Zugriffs handelt es sich nämlich zunächst um ein Wildcard, nicht einen einzelnen Verzeichnisnamen. Das bedeutet: open_basedir = "/dir/incl" schließt den Zugriff auf Verzeichnisse wie "dir/includes" und "dir/incls" mit ein. Möchten Sie den Zugriff auf ein ganz spezielles Verzeichnis beschränken, so beenden Sie die Anweisung mit einem Slash.

• PHP Sicherheits- und Safe Mode Konfigurationsanweisungen

<http://us3.php.net/manual/de/feature...#ini.safe-mode>

• Hier einige Beispielanweisungen zu den oben genannten Überlegungen:

```
PHP-Code:      register_globals = 0
disable_functions = show_source, system, shell_exec, passthru, exec, phpinfo, popen, proc_open
allow_url_fopen = 0
magic_gpc_quotes = 0
safe_mode = 1
```

open_basedir = /dir/to/include/change_me/ Joomla! und noch mehr Sicherheit Dieser Abschnitt richtet sich an alle
die auf der Suche nach der maximalen Sicherheit sind, wie beispielsweise e-Commerce Seiten. Einige der angesprochenen Punkte sollten Sie allerdings nur dann ernst nehmen, wenn Sie an mehr als der durchschnittlichen "Paranoia" leiden, überdurchschnittliche Kenntnisse haben, über ausreichend Server Kontrolle verfügen und der Bastlertyp sind, der gerne experimentiert und lernt.

Was Sie in diesem Fall also unbedingt tun sollten:

• Engagieren Sie einen Joomla! Profi, der sich mit Sicherheitsfragen auskennt, um Ihre Seitenkonfiguration überprüfen zu lassen.

• Gehen Sie nicht auf einen Shared Server. Einige Experten sind sich diesbezüglich uneins.

• Egal wie Ihr Server konfiguriert ist, stellen Sie sicher, dass nur Sie und die, denen Sie vertrauen, auf Ihren Server Zugriff haben.

• Nutzen Sie SSL Verbindungen für Login und administrative Funktionen und natürlich auch für vertrauliche Kundeninformationen und Kundendaten. Wenn Sie Ihre Kundendaten nicht sichern, sollten Sie wenigstens einen exzellenten Anwalt haben.

• Trennen Sie Entwicklungs- und Produktionsumgebung und deaktivieren bzw. entfernen Sie administrative Funktionen innerhalb des Frontends. Nutzen Sie dafür lieber Sub-Domains, diese eignen sich hervorragend dafür.

Was wir Ihnen in diesem Fall empfehlen:

• Verschieben Sie Core Dateien, die keine Schreibrechte benötigen, nach außerhalb des Webroot Verzeichnisses und ändern Sie entsprechend die Path-Variablen oder nutzen Sie Symlinks. Beispiele hierfür sind z.B. die Konfigurationsdatei oder das Administrationsverzeichnis.

• Um den direkten URL Zugriff auf Erweiterungen zu vermeiden, editieren Sie die Apache Konfiguration .htaccess oder verschieben Sie auch diese Dateien nach außerhalb Ihres Webrootverzeichnisses und ändern Sie wieder entsprechend die Path-Variablen.

Optional wäre da noch [Anmerkung: Sie müssen natürlich in diesem Fall die Anpassung der Pfade bei jedem Joomla Update erneut vornehmen]:

• Stellen Sie sicher, dass sich alle beschreibbaren Verzeichnisse außerhalb des Webroot Verzeichnisses liegen und ändern Sie wieder entsprechend die Path-Variablen. Kontrollieren Sie auch Drittanbieter Erweiterungen hinsichtlich eigener Download Verzeichnisse oder beschreibbarer Dateien.

• Legen Sie das gesamte Joomla! Verzeichnis bis auf die Dateien, die direkten httpd Zugriff benötigen, außerhalb des Webroot Verzeichnisses und ändern Sie wieder entsprechend die Path-Variablen.

• Hat das schon mal jemand ausprobiert?

• programmieren Sie Erweiterungen zur besseren Kontrolle über den Namensraum in PHP lieber objektorientiert
• Blockieren Sie erweiterungs-übergreifende Datenbank Zugriffe. Viel Glück dabei! Bitte lassen Sie uns eine gute Lösung wissen.

• Erstellen Sie eine separate Datenbank und MySQL Accounts für vertrauenswürdige Erweiterungen

• Erstellen Sie eine weitere Datenbank mit weniger Zugriffsrechten für den Zugriff durch weniger vertrauenswürdige Erweiterungen (im Grunde also alle)

• Wenn Sie den Script Kiddies nicht sagen möchten, welche URL vor unberechtigtem Zugriff gesichert werden soll, können Sie versuchen, auf eine 404 Error Seite weiterzuleiten. Dies wird ihnen anzeigen, dass die URL nicht gültig ist, oder zumindest eher als dass sie existiert, aber zusätzlich besondere Schutzmaßnahmen benötigt. (Vergessen Sie nicht, die 404 Fehlerseite zu erstellen!)

• Im offiziellen Joomla Forum stellt einer der User klar:

• Wenn zum Hacken der Seite etwas anderes als den Browser verwendet wird, so wird das Skript header() schlicht

ignorieren und der restlichen Code ausgeführt. Wenn Sie diese Umleitung wirklich haben möchten, müssen Sie eine EXIT Anweisung hinzufügen. Im Beispiel unten werden beide Methoden verwendet. Dies ist natürlich der "Overkill", aber wird sicherlich standhalten.

```
PHP-Code:      if(!defined( '_VALID_MOS' )){
                header("Location: http://www.mysite.com/404.html/");
                exit;
            }
defined( '_VALID_MOS' ) or die("Restricted access");
```

Entwickler von Erweiterungen

- Folgen Sie den Joomla! Programmierstandards (auf der offiziellen Seite forum.joomla.org/index.php/board,164.0.html, siehe dazu auch auf der Hilfeseite von Joomla.de unter www.joomla.de)
- Arbeiten Sie folgende Dokumente durch, dort lesen Sie- von Joomla! Core Entwicklern geschrieben- wie Sie eine sichere Komponente erstellen forum.joomla.org/index.php/topic,78781.0.html
- Entwickeln Sie lokal und kopieren Sie die Erweiterung dann auf einen Produktionsserver, wenn die Testphase vollständig abgeschlossen ist
- Nutzen Sie XAMPP für Ihre lokale Installation

Ihre Aufgabe:

Diese Arbeit wird ständig aktualisiert. Korrekturen und Ergänzungen werden dankend angenommen.